**Class design:**

**Types**

* **Type name**
* **Array of weaknesses**
* **Array of Resistances**

**Moves**

* **Move name**
* **Damage**
* **Accuracy**
* **Type(class)**
* **Power points**
  + **If run out use struggle**
* **Status Effects**
  + **Status effect chance**
* **Description**
* **Visual (animation)**

**Pokémon**

* **Name**
* **Level (class) increase all stats by 1**
* **Status effect (string)**
* **Type (s) (array of type)**
* **Stats (class)**
* **Ability(class)**
* **Learn Set (array of strings)**
  + **made up of move name and the last 2 indexes are lv that the move is learned at**
  + **Every level up check through this list to see if any of moves are at the same level as current level.**
* **Move Dictionary (global array of moves)**
* **Moves (array of 4 moves)**
  + **When level up check if**

**Abilities**

* **Ability Name**
* **Run appropriate Method in update**

**STAB**

**-if move is the same type as user gain 25%**

**-if move is effective against enemy type gain 25% for each type and vice versa with ineffective**

**Move Dictionary**

**Status Effects**

* **Paralysis- chance to remove Pokémon’s turn, can\_Move variable**
* **Poison- reduce health every turn if Pokémon is currently out of poke ball, and outside of battle.**
* **Burn- reduce health every turn if Pokémon is currently out of poke ball.**